Chapter

4

Trust-Region Method

4.1 Introduction

Iterative methods for optimization are categorized into two classes. One class is called line search methods and the other class as trust region algorithms. Trust-Region methods are iterative method in which a model (m_k) approximates the objective function (f) and this model is minimized in a neighborhood of the current iterate (the trust region). In case of a line-search method, the iterations are performed toward some particular directions; for example, the gradient directions are used to find the successive iterates in *steepest descent* [Liu and Chen, 2004]. However, in a Trust-Region algorithm, its iterates are derived by solving the corresponding optimization problem iteratively within an enclosed region. Therefore, we have more options to choose the iterates. Indeed, we can consider line-search methods as special cases of trust region methods [A. R. Conn *et al.*, (2000)]. Trust-Region methods first introduced by M. J. D. Powel in 1970 [M. J. D. Powell, (1970)]. Powell [M. J. D. Powell, (1975)] also established the convergence result of unconstrained Trust-Region method optimization. Fletcher [R. Fletcher, (1972)] first recommended Trust-Region algorithms to solve linearly constrained optimization problems and non-smooth optimization problems [R. Fletcher, (1982)]. Trust-Region methods are very essential and effective methods in the area of nonlinear optimization. These methods are also useful for non-convex optimization problems and non-smooth optimization problems [Sun (2004)].

As most of the research works on trust region algorithms are mostly started in the 80s, trust region algorithms are less mature compare to line search algorithms, and the applications of trust region algorithms are limited as compared to line search algorithms. However, trust region methods have two major advantages. One is that they are reliable and robust; another is that they have very strong convergence properties. The key contents of any trust region algorithm are how to calculate the trust region trial step and how a decision can be made if a trial step should be accepted or not. An iteration of a trust region algorithm has the following form; a trust region is available at the beginning of the iteration. This is possible by considering an initial guess value $X_0 \in \mathbb{R}^n$ and trust

 https://doi.org/10.1017/CBO9781316134504.005

region radius $\Delta_0 > 0$. An approximate model (m_k) is constructed, and it is solved within the trust region, giving a solution s_k which is called the trial step. A merit function is selected, which is used for updating the next trust region and for deciding the new iterate point [Yuan, (1993)]. There are many applications of Trust-Region method like curve fitting (Helfrich and Zwick, 1996), optimization of pressure swing adsorption [Agarwal *et al.* 2009] etc.

4.2 Basic Trust-Region Method

4.2.1 Problem statement

The statement of Basic Trust-Region Method is given below

i. consider a local model m_k of the objective function f around X_k

ii. calculate a trial point $X_k + s_k$ that reduces the model m_k within the trust region $\|s_k\| \le \Delta_k$

iii. calculate the reduction ratio

$$r_k = \frac{f(X_k) - f(X_k + s_k)}{m_k(X_k) - m_k(X_k + s_k)} \tag{4.1}$$

iv. if m_k and f agree at $X_k + s_k$, i.e., $r_k \ge \eta_1$

then

accept trial point: $X_{k+1} = X_k + s_k$ (4.2)

update the trust region radius:

$$\Delta_{k+1} \in \begin{cases} [\Delta_k, \infty] & \text{if} \quad r_k \ge \eta_2 \\ [\gamma_2 \Delta_k, \Delta_k] & \text{if} \quad r_k \in (\eta_1, \eta_2) \end{cases} \tag{4.3}$$

else

reject trial point: $X_{k+1} = X_k$

decrease trust region radius: $\Delta_{k+1} \in [\gamma_1 \Delta_k, \gamma_2 \Delta_k]$

where $0 < \eta_1 \le \eta_2 < 1$ and $0 < \gamma_1 \le \gamma_2 < 1$

The reduction ratio r_k plays two roles in BTR method

1. decides the acceptance of the trial point $X_k + s_k$
2. controls the update procedure of Trust-Region radius

Unlike the line search methods, Trust-Region methods will attempt to get the next approximate solution within a region of the current iterate. For trust region methods, three issues need to be emphasized: (i) Trust-Region radius, which determines the size of a trust region (ii) Trust-Region subproblem, which approximates a minimizer within the region and (iii) Trust-Region fidelity;

used to estimate the level of accuracy of an approximated solution [Liu and Chen, 2004]. The following sections describe Trust-Region radius, Trust-Region subproblem, and Trust-Region fidelity in depth.

4.2.2 Trust-Region radius

At kth iteration, the quadratic model of the function $f(X)$ around the current iterate X_k is represented by

$$m_k\left(X_k + d\right) = f\left(X_k\right) + g_k^T d + \frac{1}{2} d^T B_k d, \tag{4.4}$$

The Trust-Region is characterized as the region where $\|d\| \le \Delta_k$.

$$\Delta_{k+1} = \beta_k \Delta_k \tag{4.5a}$$

$$\beta_k \in \begin{cases} [\gamma_1, 1] & \text{if } r_k < \mu_1 \\ [1, \gamma_2] & \text{if } r_k \ge \mu_2 \\ [\gamma_3, \gamma_4] & \text{otherwise} \end{cases} \tag{4.5b}$$

where Δ_k is a trust region radius (Fig. 4.1). The constants are as follows $0 < \gamma_1 < 1 < \gamma_2$, $\gamma_1 \le \gamma_3 < 1 < \gamma_4 \le \gamma_2$ and $0 < \mu_1 < \mu_2 < 1$. A function called merit function is generally used to check whether the trial step is accepted, or any adjustment is needed for the trust region radius. Choosing a proper trust region radius (Δ_k) is crucial for any TR method algorithm. A trust region may show an improvement at very slow rate during the estimation of the solution, if the value of Δ_k is very small. On the other hand, the agreement between the model (m_k) and the objective function (f) will be very poor when the value of Δ_k is too large. At the initial step, selection of an initial trust region radius (Δ_0) is very important. During the implementation of a trust region method, we should be concerned regarding the issue of the proper selection of the initial trust region radius (ITRR) [M. J. D. Powell, (1970)]. A bad choice of Δ_0 can lead an increase in number of iteration consequently cost of optimization, particularly when the linear algebra required for each iteration is costly [A. SARTENAER, (1997)]. Now, we will discuss an algorithm for finding initial Trust-Region radius.

Initial Trust Region Radius (ITRR)

The initial trust region radius (Δ_0) can be estimated automatically by following a strategy that is introduced by Sartenaer [A. Sartenaer, (1997)]. The fundamental concept of this method is to find out a maximal initial radius by performing several repeated trials in the $-g_k$ direction that also guarantee a satisfactory agreement between the objective function and the model. The difficulty in estimating an ITRR (Δ_0) is to find a technique to check agreement between the objective function and the approximated model at the initial point X_0. The technique discussed in this chapter is based on the utilization of normally available information at this point, i.e., value of the function and its gradient. A reliable ITRR can be determined with the additional cost of some function evaluations.

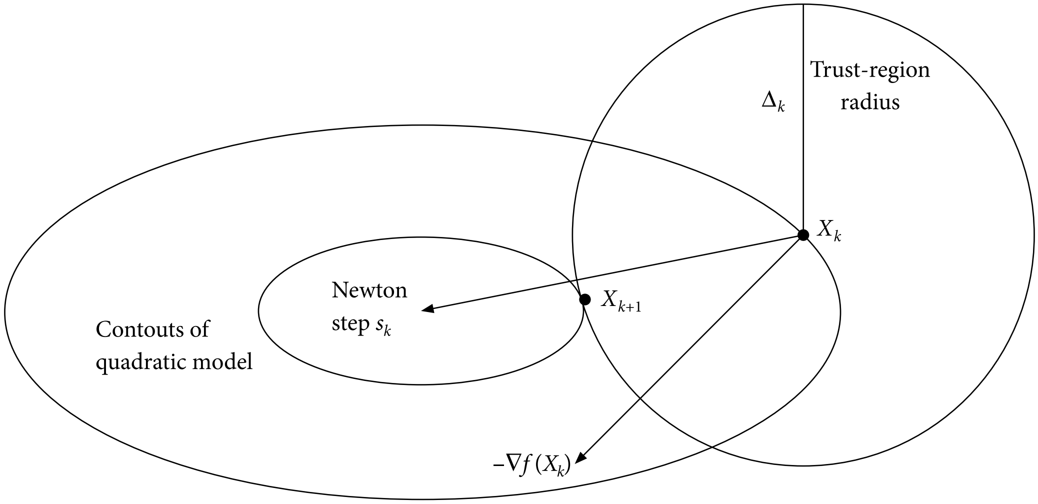

Fig. 4.1 Trust region

We need to perform some extra mathematical computations for this purpose. However, this technique will reduce the number of iterations required to obtain a solution.

This method determines the maximal radius, which shows a satisfactory agreement between the objective function and the model in the $-g_0$ direction, employing an iterative search procedure along this direction. At any iteration k of the search procedure, with a given radius estimation $\Delta_{k,0}$, the values of the model and objective function are calculated at the point $\left(X_0 - \Delta_{k,0}\hat{g}_0\right)$, is

$$m_{k,0} = m_0\left(X_0 - \Delta_{k,0}\hat{g}_0\right) \tag{4.6a}$$

and
$$f_{k,0} = f_0\left(X_0 - \Delta_{k,0}\hat{g}_0\right) \tag{4.6b}$$

where
$$\hat{g}_0 = \left(\frac{g_0}{\|g_0\|}\right) \tag{4.6c}$$

And the ratio, $$r_{k,0} = \frac{f_0 - f_{k,0}}{f_0 - m_{k,0}} \tag{4.7}$$

is calculated, then the algorithm stores the maximum value among the estimates of $\Delta_{k,0}$, whose related $r_{k,0}$ is "close enough to one". Finally, it updates the current estimate $\Delta_{k,0}$. The updating phase for $\Delta_{k,0}$ follows the framework in Eq. (4.5a)–(4.5b), however, includes a more general test on $r_{k,0}$ because the predicted change in Eq. (4.7) is not guaranteed to be positive. Therefore, we set

$$\Delta_{k+1,0} = \beta_{k,0}\Delta_{k,0}$$

where

$$\beta_{k,0} \in \begin{cases} [\gamma_1, 1] & \text{if } |r_{k,0} - 1| > \mu_1 \\ [1, \gamma_2] & \text{if } |r_{k,0} - 1| \le \mu_2 \\ [\gamma_3, \gamma_4] & \text{otherwise} \end{cases} \tag{4.8}$$

for some $0 \le \mu_2 < \mu_1$. Note that updated Eq. (4.8) only considers the tolerability between the objective function and its approximated model, without taking care of the minimization of the objective function f.

4.2.3 Trust-Region subproblem

Trust region subproblems are one of the vital parts of trust region algorithms. Since, each iteration of a trust region algorithm requires to solve (exactly or inexactly) a trust region subproblem, finding efficient solver for trust region subproblems is very important. We consider subproblem (4.4) which has been studied by many authors. The following lemma is well known for solving Trust-Region subproblem.

Lemma 4.1

A vector $d^* \in \Re^n$ is a solution of the problem

$$\min_{d \in \Re^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d, \tag{4.9}$$

subject to $\|d\| \le \Delta_k$ (4.10)

where $g_k \in \Re^n$, $B_k \in \Re^{n \times n}$ is a symmetric matrix, and $\Delta_k > 0$, if and only if there exists $\lambda^* \ge 0$ such that

$$(B_k + \lambda^* I) d^* = -g_k \tag{4.11}$$

and that $B_k + \lambda^* I$ is positive semi-definite, $\|d^*\| \le \Delta_k$ and $\lambda^* (\Delta_k - \|d^*\|) = 0$

Similar to other algorithms, performance of Trust-Region methods depend on the selection of a set of parameters.

4.2.4 Trust-Region fidelity

When a subproblem is solved, the trial point $X_k + s_k$ will be checked to understand whether it is a good candidate for the next iterate. This is calculated explicitly using the equation

$$r_k = \frac{f(X_k) - f(X_k + s_k)}{m_k(X_k) - m_k(X_k + s_k)} \tag{4.12}$$

The trial point is accepted whenever $r_k \geq \eta_1$, i.e., $X_{k+1} = X_k + s_k$. If not, then $X_{k+1} = X_k$. As the parameter η_1 is a small positive number, the aforesaid rule favors a trial point only if the value of the objective function f is also decreased. The radius of the Trust-Region will be expanded for the next iteration while m_k approximates f nicely and gives a large value of r_k. Conversely, when the value of r_k is smaller than η_1 or r_k is negative, it implies that within this present trust region the objective function f is not approximated properly by the model m_k. As a result, the iterate remains unchanged. Then, the Trust-Region radius will be reduced in size to develop more suitable model and subproblem for the next iteration.

4.3 Trust-Region Methods for Unconstrained Optimization

During the minimization of an unconstrained problem, a step to a new iterate is estimated by minimizing a local model of the objective function within a bounded region that has the center at the current iterate. At every iteration of the trust region method, the nonlinear objective function is substituted by a simple model keeping its center at the current iterate. For this purpose, the first and probably second-order information available at this present iterate is used to construct this simple model. Therefore, this model is usually appropriate only within a certain restricted region neighboring this point. Thus, a trust region is defined where the approximated model is said to agree satisfactorily with the actual objective function. Then the trust region algorithms are composed of solving a series of subproblems in which the model is minimized approximately within the trust region, which gives a candidate for the next iterate [M. J. D. Powell, (1970)].

In this section, we consider trust region algorithms for unconstrained optimization problem:

$$\min_{X \in \Re^n} f(X) \tag{4.13}$$

where $f(X)$ is a nonlinear continuous differentiable function in $\Re^n$. At each iteration, a trial step is evaluated by solving the subproblem

$$\min_{d \in \Re^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d \tag{4.14}$$

$$\text{subject to } \|d\| \leq \Delta_k \tag{4.15}$$

where $g_k = \nabla f(X)$ is the gradient at the current approximate solution, B_k is an $n \times n$ symmetric matrix that approximates the Hessian matrix of $f(X)$ and $\Delta_k > 0$ is called the trust region radius. Let s_k be a solution of (4.14)–(4.15). The predicted reduction is defined by the reduction in the approximate model, that is

$$P_{\mathrm{red}_k} = m_k(0) - m_k(s_k) \tag{4.16}$$

Unless the current point X_k is a stationary point and B_k is positive semi-definite, the predicted reduction P_{red_k} is always positive. The actual reduction is the reduction in the objective function:

$$A_{\text{red}_k} = f\left(X_k\right) - f\left(X_k + s_k\right) \tag{4.17}$$

And we define the ratio between the actual reduction and the predicted reduction by

$$r_k = \frac{A_{\text{red}_k}}{P_{\text{red}_k}} \tag{4.18}$$

which is employed to make a decision if the trial step is acceptable and to adjust the new trust region radius.

A general algorithm for unconstrained trust region optimization can be given as follows.

Step 1 Given $X_1 \in \Re^n$, $\Delta_1 > 0$, $\varepsilon \geq 0$, $B_1 \in \Re^{n\times n}$ symmetric; $0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 \leq \tau_0 \leq \tau_2 < 1$, $\tau_2 > 0$, $k = 1$

Step 2 If $\|g_k\| \leq \varepsilon$ then stop; Solve a trust region subproblem, giving s_k.

Step 3 Compute $r_k = \dfrac{P_{\text{red}_k}}{A_{\text{red}_k}}$; set $X_{k+1} = \begin{cases} X_k & \text{if } r_k \leq \tau_0 \\ X_k + s_k & \text{otherwise} \end{cases}$

choose Δ_{k+1} from the equation $\Delta_{k+1} = \begin{cases} \left[\tau_3 \|s_k\|,\ \tau_4 \Delta_k\right] & \text{if } r_k < \tau_2 \\ \left[\Delta_k,\ \tau_1 \Delta_k\right] & \text{otherwise} \end{cases}$

Step 4 Update B_{k+1}; $k = k + 1$; go to step 2

The constants τ_i $\left(i = 0,1,\ldots,4\right)$ can be selected by users. Typical values are $\tau_0 = 0$, $\tau_1 = 2$, $\tau_2 = \tau_3 = 0.25$, $\tau_4 = 0.5$

4.4 Trust-Region Methods for Constrained Optimization

For constrained problems, most trust region subproblems can be regarded as some kind of modification of the SQP subproblem of line search algorithm, which has the following form:

$$\min_{d \in \Re^n} m_k\left(d\right) = g_k^T d + \frac{1}{2} d^T B_k d, \tag{4.19}$$

subject to

$$c_i\left(X\right) + d^T \nabla c_i\left(X\right) = 0\,; \quad i = 1,2\ldots,m_e \tag{4.20}$$

$$c_i(X) + d^T \nabla c_i(X) \geq 0\,; \quad i = m_e + 1, \ldots, m \tag{4.21}$$

where $g_k = g(X_k) = \nabla f(X_k)$ and B_k is an approximate Hessian of the Lagrange function. The first type of trust region subproblems, being a slightly modification of SQP subproblem (4.19)–(4.21), have the following form:

$$\min_{d \in \Re^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d, \tag{4.22}$$

subject to

$$\theta_k c_i(X) + d^T \nabla c_i(X) = 0\,; \quad i = 1, 2 \ldots, m_e \tag{4.23}$$

$$\theta_k c_i(X) + d^T \nabla c_i(X) \geq 0\,; \quad i = m_e + 1, \ldots, m \tag{4.24}$$

$$\|d\| \leq \Delta_k \tag{4.25}$$

where $\theta_k \in (0,1)$ is a parameter that is introduce to overcome the possible non-feasibility of the linearized constraints (4.20) and (4.21) in the trust region (4.25). Another trust region subproblem is obtained by substituting the linearized constraints (4.20) and (4.21) by a single quadratic constraint. It can be written as:

$$\min_{d \in \Re^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d, \tag{4.26}$$

subject to

$$\left\| \left(c_k + A_k^T d \right)^- \right\| \leq \xi_k \tag{4.27}$$

$$\|d\| \leq \Delta_k \tag{4.28}$$

where $c_k = c(X) = \left(c_1(X), \ldots, c_m(X)\right)^T$, $A_k = A(X) = \nabla c(X)^T$, $\xi_k \geq 0$ is a parameter and the superscript "–" means that $v_i^- = v_i \left(i = 1, 2, \ldots, m_e\right)$, $v_i^- = \min\left[0, v_i\right]\left(i = m_e + 1, \ldots, m\right)$. This algorithm is given by Celis *et al.* [M. R. Celis *et al.,* (1985)] and Powell and Yuan [Powell and Yuan, (1991)]

Trust region subproblems can also derived by using exact penalty functions. The following trust region subproblem is based on the L_∞ exact penalty function:

$$\min_{d \in \Re^n} m_k(d) = g_k^T d + \frac{1}{2} d^T B_k d + \sigma \left\| \left(c_k + A_k^T d \right)^- \right\|_\infty \tag{4.29}$$

$$\text{subject to } \|d\| \le \Delta_k \tag{4.30}$$

Trust region subproblems based on exact penalty functions are closely related to subproblems of trust region algorithms for nonlinear systems of equations. Trust region algorithms that compute the trial step by solving (4.29)–(4.30) are also similar to trust region algorithms for nonsmooth optimization.

Once, a trial step s_k is computed by solving the trust region subproblem, the predicted reduction P_{red_k} is defined by the reduction of some approximate function $\overline{m}_k(d)$. It should be noted that in general $m_k(d) - \overline{m}_k(d)$. A merit function $P_k(X)$ is used to define the actual reduction A_{red_k}. $P_k(X)$ is normally some penalty function. And the functions $\overline{m}_k(d)$ and $P_k(X)$ are so constructed that

$$m_k(d) - \overline{m}_k(0) = P_k(X+d) - P_k(X) + O(\|d\|) \tag{4.31}$$

when $\|d\|$ is very small.

The algorithm can be stated as follows:

Algorihtm

Step 1 Given $X_1 \in \Re^n$, $\Delta_1 > 0$, $\varepsilon \ge 0$, $B_1 \in \Re^{n\times n}$ symmetric; $0 < \tau_3 < \tau_4 < 1 < \tau_1$, $0 \le \tau_0 \le \tau_2 < 1$, $\tau_2 > 0$, $k = 1$

Step 2 If $\|g_k\| \le \varepsilon$ then stop; Solve a trust region subproblem, giving s_k.

Step 3 Compute $r_k = \dfrac{P_{\text{red}_k}}{A_{\text{red}_k}}$; set $X_{k+1} = \begin{cases} X_k & \text{if } r_k \le \tau_0 \\ X_k + s_k & \text{otherwise} \end{cases}$

choose Δ_{k+1} from the equation $\Delta_{k+1} = \begin{cases} \left[\tau_3 \|s_k\|,\ \tau_4 \Delta_k\right] & \text{if } r_k < \tau_2 \\ \left[\Delta_k,\ \tau_1 \Delta_k\right] & \text{otherwise} \end{cases}$

Step 4 Update B_{k+1}; $k = k + 1$; go to step 2

The constants $\tau_1\,(i = 0,1,\ldots,4)$ can be chosen by users. Typical values are $\tau_0 = 0$, $\tau_1 = 2$, $\tau_2 = \tau_3 = 0.25$, $\tau_4 = 0.5$.

The parameter τ_0 is usually zero or a small positive constant. There is an advantage of using zero τ_0; whenever the objective function value is decreased, the trial step is accepted. Therefore, it would not discard a “good point”, which is a desirable property particularly when the evaluations of functions are very costly.

4.5 Combining with Other Techniques

Combination of Trust-Region method with other techniques is also possible in constructing algorithms. Nocedal and Yuan [Nocedal and Yuan, (1998)] observed that the trial step of trust region algorithms for unconstrained optimization is also a descent direction. Therefore, when the trial step is unacceptable, it is still possible to carry out a line search along the direction of the trial step. An algorithm, which combines backtracking and trust region has been discussed by Nocedal and Yuan.

4.6 Termination Criteria

Convergence of a Trust-Region algorithm relies on the accuracy of the approximate solution to the Trust-Region subproblem (4.9). For an efficient algorithm, it is important that the Trust-Region subproblem not be solved exactly. Broadly speaking, it is desirable to compute a solution with as little effort as possible, subject to the requirement that the computed solution does not interfere with the overall convergence of the method.

Fletcher (1987) discussed some termination criteria. The conventional termination criteria are to terminate when any of $f(x_k) - f(x_{k+1})$, $x_k - x_{k+1}$, or $\nabla f(x_k)$ are small. All are used in practice. Fletcher also suggested

$$\left(x_k - x_{k+1}\right)^T \nabla f\left(x_k\right) \tag{4.32}$$

which is invariant when the step is a Newton step.

It is also possible for $\Delta_k \to 0$ as $k \to \infty$. Thus, it is also necessary to consider termination when Δ_k is small

i. The change in the objective function value $|f(x) - f(x_{k+1})|$ is less than the termination criterion ε_1 in any step

ii. The change in the second-order Taylor series model of the objective function value $\left|\left(x_k - x_{k+1}\right)^T \left(g_k + \frac{1}{2} B_k \left(x_k - x_{k+1}\right)\right)\right|$ is less than the termination criterion ε_3 in any step

iii. The trust region radius Δ_k has shrunk to less than the termination criterion ε_4 in any step.

4.7 Comparison of Trust-Region and Line-Search

For trust region method, we have employed an approximated quadratic model m_k for the implementation. If we consider first two terms in RHS of Eq. (4.4), it will be reduced to a linear model. This shows that a Trust-Region method with an approximated linear model is similar to gradient descent, however, often it attains better performances due to its capability to regulate trust regions adaptively all through the iterations. For this reason, the line-search methods can be regarded as special cases of Trust-Region [Liu and Chen, 2004]. Both the Trust-Region and the line-search methods are guaranteed to converge to a local minimum. However, for real application, all these local minima are not significant. A typical line-search method (e.g., steepest descent or even Trust-Region with a linear approximation) may often converge to a local minimum, which is substandard compared to a nearby one.

Summary

- Trust-Region methods have been discussed for both unconstrained and constrained optimization problems. Various features of Trust-Region methods such as trust region radius, trust region subproblem, trust region fidelity are given in this chapter. Termination criteria are an important factor for any algorithm; it has been considered in this chapter. Various researches confirm that the trust region methods can be combined with other optimization methods. A comparison between line search and trust region methods is made to explain the advantages and disadvantages of the methods.

Exercise

4.1 Find the quadratic approximation of the function around the point (1, 1)

$R(C, t) = 77.92 + 9.41C + 3.86t - 3.53C^2 - 7.33t^2$

4.2 What are the advantages of trust region methods over line search methods?

4.3 Discuss the effect of initial trust region radius (ITRR) on the performance of TR algorithm.

4.4 Define trust region fidelity.

4.5 Find the maximum of the function using Trust region method

$f = 15x_1 + 8x_1x_2 + 5x_2$ subject to $x_1 + x_2 \leq 10$

4.6 Calculate the reduction ratio $r_k = \dfrac{f(X_k) - f(X_k + s_k)}{m_k(X_k) - m_k(X_k + s_k)}$ for the function given in problem 1.

4.7 Is the result of any Trust Region method depends on the initial guess value $X_0 \in \mathbb{R}^n$? Justify your answer.

4.8 Discuss the convergence criteria of Trust Region method, show how the other parameters (τ_1, τ_2, τ_3 and τ_4) affect the convergence.

4.9 When combination of other methods with Trust region method is advantageous?

4.10 Write an algorithm for problem 5, which combines Trust region and backtracking method.

References

Agarwal, A., Biegler, L. T. and Zitney, S. E. 2009. *Simulation and Optimization of Pressure Swing Adsorption Systems Using Reduced-Order Modeling;* Ind. Eng. Chem. Res., 48, 2327–43.

Celis, M. R., Dennis, J. E. and Tapia, R. A. *A Trust Region Algorithm for Nonlinear Equality Constrained Optimization,* in: P. T. Boggs, R. H. Byrd and R. B. Schnabel, eds., Numerical Optimization (SIAM, Philadelphia, 1985), 71–82.

Conn, A. R., Gould, N. I. M. and Toint, P. L. 2000. *Trust-Region Methods,* SIAM.

Fletcher, R. An *Algorithm for Solving Linearly Constrained Optimization Problems,* Math. Program. 2(1972) 133–65.

Fletcher, R. *A Model Algorithm for Composite Non-differentiable Optimization Problems,* Mathematical Programming Study, 17(1982): 67–76.

Fletcher, R. 1987. *Practical Methods of Optimization* (2*e*), John Wiley and Sons.

Helfrich, H. P. and Zwick, D. *A Trust Region Algorithm for Parametric Curve and Surface Fitting,* Journal of Computational and Applied Mathematics, 73(1996): 119–34.

Liu, T-L, Chen, H-T. 2004. *Real-Time Tracking Using Trust-Region Methods*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 26, No. 3.

Nocedal, J. and Yuan, Y. *Combining Trust Region and Line Search Techniques,* in: Y. Yuan, ed. Advances in Nonlinear Programming, (Kluwer, 1998), 153–75.

Powell, M. J. D. 1970. *A New Algorithm for Unconstrained Optimization, in Nonlinear Programming,* J. B. Rosen, O. L. Mangasarian, and K. Ritter, eds., New York: Academic Press.

Powell, M. J. D. 1975. *Convergence Properties of a Class of Minimization Algorithms,* in: O. L. Mangasarian, R. R. Meyer, S. M. Robinson (Eds.), Nonlinear Programming 2, New York: Academic Press.

Powell, M. J. D. and Yuan, Y. *A Trust Region Algorithm for Equality Constrained Optimization,* Math. Prog. 49(1991): 189–211.

Sartenaer, A. 1997. *Automatic Determination of an Initial Trust Region in Nonlinear Programming,* SIAM J. Sci. Comput., 18(6): 1788–1803.

Sun, W.; Appl. Math. Comput., 156(2004): 159–74.

Yuan, Y., *A New Trust Region Algorithm for Nonlinear Optimization,* in: D. Bainov and V. Covachev, eds., Proceedings of the First International Colloquium on Numerical Analysis (VSP, Zeist, 1993), pp. 141–52.